

SyncWare News

The journal for technical
applications of T/S
computers

Volume 5 Number 1 Sept.-Oct. '87

Happy 5th!

Believe it or not, here we are about to embark on our fifth year of publishing SYNCWARE NEWS. We are committed to the continued technical, hardware, and software support of Timex TS1000/ZX81, TS1500, and TS2068 computers. For the next year, we're going to guarantee to bring you five more issues packed with ideas, tips, and quips regarding our favorite "little" computers.

As we continue with Volume #5, we could use a little help from our readers. If you're reading SyncWare News, benefiting from it, but you're not a subscriber, then please subscribe. If you have an idea for an article, submit it. If you have constructive criticism concerning the contents or appearance of this magazine, tell us. This magazine exists because we feel there is a need for it, and you fine folks have continued to support it. We sincerely appreciate your support and highly value your input. Oh, just one more thing. The next time you hoist a cup of your favorite brew (vintage Pepsi for me-ed.), join us in toasting SyncWare News' Fifth Anniversary.

They Like the Z88

The September 1987 issue of Great Britain's prestigious Personal Computer World magazine contains a very favorable review of the Cambridge Computer Z88 Laptop Computer. The reviewer, Robert Schifreen, felt that the machine was a good value (approx. \$475.00 US), provided five times the computing power of the Tandy 100 at half the weight, and has excellent bundled software. On the minus side were flimsy physical construction, a poor space bar, and the inability to send and receive files via the bundled terminal emulation software. Overall the reviewer liked the machine and felt owning one had changed his life for the better.

And in This Corner

Tom Woods, the venerable publisher of SyncWare News, will be stepping down from that post in the near future. An open letter to our readers from Tom appears in this issue's Forum column (see Changing of the Guard). Current SyncWare News Editor, Jeff Moore, will be assuming the duties of publisher, the keeper of subscriptions, and bookkeeper. We are pleased to announce that our own Basil Wentworth will be taking on the mantel (cleaned, pressed, one size-fits-all) of Editor. Basil's address is 1413 Elliston Drive, Bloomington, IN 47401. Tom Bent will remain as Editor of Quantum Levels and Fred Nachbaur will continue as the Timex Guru.

FOR YOUR SUPPORT

TS-2068 SAFE DISK UPDATE, 1317 Stratford Ave., Panama City, FL 32404, (904) 871-4513, is a quarterly newsletter devoted to the support of users who have upgraded to disk drive systems. A section features the Oliger Safe Disk Operating System (SDOS). The annual subscription is \$12.

CHARLES STELDING, 1415 South Baxter, Tyler, TX 75701, brings you TIMEX 2068 DESK TOP PUBLISHER. This new program allows the user to layout pages for bulletins, reports, announcements, newsletters, etc. The program will generate headlines, text, and allow placement of SCREEN's anywhere on the page. The user may also design his own fonts to add to the ones already included. For the 2040 and most dot matrix printers with appropriate interface. Price \$19.95 plus S&H, Olivetti 2300 version \$24.95 plus S&H.

SHAWN STAMP, 3405 Mark Lane, Norton, OH 44203, (216)825-8957, challenges you to see just what your TS2068 is capable of doing! CHOPPER COMMAND is billed as a fast-paced action game that takes over where Rambo left off. As a chopper pilot in the Air Force, you rescue POW's from the jungles of Vietnam. Comes on cassette with instructions to transfer the program to your disk system for only \$17.50 (includes S&H).

CHIA-CHI CHAO, 73 Sullivan Drive, Moraga, CA 94556-1209, has his latest catalog available. Included in its pages are bargains like Diamond Mike II and the Great Game and Graphics Show, both for only \$10. Compass, the 2068 BASIC compiler/Z80 assembler program for only \$10. Word Twister and Vocabuilder for only \$10 and Money Machine II for only \$11. Most programs are available on either cassette or Aerco disk for the TS2068.

ROBERT D. HARTUNG, 2416 No. County Line Road, Hometown, IN 46748, (219) 6373081, has offered to place orders for Beta Basic 4.0 through the British supplier for pre-paid US customers. BETA BASIC 4.0 allows structured programming and procedures on Spectrums of Spectrum-emulated computers by 100 new commands and functions to the machines. The cost is \$26 plus \$1.50 for handling in US funds by check or MO. Contact Mr. Hartung for more details.

C. W. ASSOCIATES, 419 N. Johnson St., Ada, OH 45810, (419)634-4874, are full line QL dealers. Contact them for your QL software and hardware needs. Items listed in the current flyer include an Expansion ROM board, Front Page, Better BASIC, Task Master, and games like Knight Flight, Karate, Squadrons, and Tank Blaster. MC/VISA now accepted.

E. A. BROWN COMPANY, 3404 Pawnee Drive, Alexandria, MN 56308, (612)7638847, is currently distributing their latest catalog. Hardware and software items can be found for Timex, Atari-ST, and IBM computers and compatibles. Some bargains of interest to Timex computer enthusiasts are full size printers, 2068 Mikrodrives, and TS1000 and 2068 software.

LARKEN ELECTRONICS, RR#2 Navan Ontario, Canada K4B-1H9, is currently marketing the LKDOS Cartridge. LKDOS is a replacement DOS for Aerco, Ramex, Oliger, and Larken Disc Systems, and claims to allow compatibility between the four popular TS2068 disk IF systems. LKDOS supports all the token commands such as LOAD, SAVE, CAT, ERASE, MERGE, OPEN, CLOSE, etc., and files types can be BASIC, code, or array. The DOS claims to be fully Spectrum and OS-64 compatible and supports NMI snap-shot memory saves when used in conjunction with the necessary hardware. Priced from \$65.00.

SILICON MOUNTAIN COMPUTERS, C-12 MTN. STN. GROUP BOX, NELSON, BC V1L 5P1 CANADA announces that their SCRAM board is in stock and ready for delivery. This functional equivalent and improvement of the "Hunter" board will directly run all of Silicon Mountain's high-res software. Price is \$39.95 postpaid, fully assembled and tested. Bare boards available for \$12.

SyncWare

Thomas B. Woods
Jeffrey D. Moore
Fred A. Nachbaur
Thomas J. Bent

Publisher
Editor
Technical Director
Asst. Editor

All subscription information, billing, and ad artwork should be addressed to:

SyncWare News
PO Box 64
Jefferson, NH 03583
(603) 586-7734

Article submissions, Forum, Support listings, Classified ads, and really easy questions should be addressed to:

SyncWare News
603 S. Mill St.
Louisville, OH 44641
(216) 875-1257

Article submissions, hardware, and everything else, especially the tough questions, should be addressed to:

SyncWare News
C-12, Mtn. Sta. Group box
Nelson, BC V1L 5P1
(604) 354-3858

Back issues of SWN are available for \$3.50 each. Volume #1 (for the 1000) is available for \$16.95.

SWN is done entirely on TS computers. Submissions are preferred as word processed text files recorded on good quality tape. Memotext for the 1000 and Mscript for the 2068 are the preferred word processors. Documents submitted in Tasword Two and other word processors must be done in 52 column format. If you do not have a word processor, but you feel you have a good idea, then don't hesitate too work it up and submit it. Complete writers guidelines are available upon request from the Ohio Editorial office. SWN pays authors about \$40.00 per page for original articles.

FORUM

Changes at SWN

Tom Woods

There are a few changes coming to SyncWare News that we want to tell you about. As most of you probably know, I have been publishing the "News" for quite some time--three years to be exact--and I am getting tired. To preserve a lot of things, not the least of which includes a long healthy future for the newsletter AND my sanity, I am going to take a break. I am stepping back, releasing the reins, giving myself a much needed change of scenery, etc. etc.

Jeff Moore, our editor and the person we can all thank for drumming up the fantastic mix of articles which appears each issue, has received a promotion. He will take over my responsibilities as publisher. But don't worry about him letting the job of editing the "News" go astray. Through the most extraordinary luck, we have found the perfect person to take over as editor: who else, but Basil Wentworth. I wanted Basil to write this column but when I called him two days before press time his wife Jocelyn said he was off having his head examined and wouldn't be able to get it in on time so you'll have to bear with me while I tell you what it all means.

What it Means

First of all, it means we've got fresh blood and that's good. Not because we like blood, but because we like freshness. Under Basil's able editorship, we'll continue with all the good stuff we've seen in the past, and Basil's new perspectives can only enhance our quality even more. The only thing we'll have to watch out for is Basil's penchant for puns. If he's not careful, he'll turn SyncWare News into a journal of North American humor and we'll all split a gut from laughing so hard and forget about our computers. But seriously, SyncWare couldn't have found a better editor. The first time I stumbled into Basil was when he purchased my old ZX Pro/File program and he sent me a list of grammatical errors he found in the manual. Problem was, his list was longer than the book! Since that first encounter, we have shared many experiences. I am proud to call him a friend, and I know SyncWare will benefit greatly from his knowledge of both writing AND computers.

Secondly, it means that Jeff Moore, as new head honcho, will have his hands full while he learns how to take people's subscription money (that was my job). Don't worry, he's honest to a fault. Jeff has many things going for him as he takes the reins. In my list, he comes second only to my wife, Adele, in the "organizational skills" department. The lack thereof has been the greatest of my many flaws. With Jeff in charge, we can all expect to see every issue come out on time with no mix-ups. Well, perhaps before we stick our necks out too far, we should revise that last statement: We will try to have every issue come out no later than.... After all, we have a tradition established. Just because we're going through a few minor changes doesn't mean we have to get radical!

Speaking of not changing, SyncWare's two ace technical wizards, Fred Nachbaur and Tom Bent, are remaining in the same capacity. No changes there.

Now the important part: where to send your money for subscriptions, renewals, back issues, and charitable contributions: All business is now conducted from the Louisville office. Orders and change of address notices should be sent to:

Jeff Moore, Publisher
SyncWare News
602 South Mill Street
Louisville, OH 44641

Please read the above paragraph again. It is very important. I realize that a few of you don't actually read the "News" and just like to send your money in so you can look at the pictures. We will be changing our address at the post office so mail will be properly forwarded, but it may cause delays, and Lord knows, we don't need any more of them! So write to Jeff. Give him a pat on the back and a few words of encouragement. And tell him what you like and dislike about SyncWare News. He needs to know.

As for myself, I am still playing with my computer and certainly still writing. You have definitely not seen the last of me. But I have been at the focal point of SyncWare for long enough. It's time to step down and let somebody else have a turn. We are indeed fortunate to have Jeff and Basil's talent available to carry on.

The US Snail

Recently we received a letter from a fellow who felt our Technical Editor, Fred Nachbaur, had given him the run-around, the cold shoulder, and ripped him off. The writer felt that he hadn't gotten a response as quickly as he thought he should. The fact of the matter was, their correspondence had crossed in the mail.

The reason we're bringing this up is to remind our readers, especially those not living in the US, or as in this case, someone in the States writing to Fred in Canada, the Postal Service can be very slow when it comes to international mail. As an example, on several occasions Fred has sent packages to the Editorial Offices in Louisville, Ohio, that have taken more than six weeks to get here. Neither of us can explain the problem, but we suspect that Customs Agents on one or both sides of the border have enjoyed playing with our software.-ed.

MIDI Data

An article written by D. M. Gualtieri was published in the "Computer Music Journal", Vol. 10 No. 3, describing the hardware for an output-only interface between a Centronics printer port and a MIDI network. The interface was described as using commonly available ICs and was estimated to cost around \$25 to build. Back issues of "CMJ" can be purchased for \$8.50 from MIT Press - Journals Dept., 55 Hayward St., Cambridge, MA 02142.

On the software side (the above deals only with the hardware), R. Moog published an excellent exposition of MIDI codes in the "Journal of the Audio Engineering Society" (vol. 34 no. 5). A reprint of this article is available from The Audio Engineering Society, 60 E. 42 St., New York, NY 10165-0075, for \$3.

Dear Editor, I would be interested in seeing as much published from the British assembler routines as possible in SyncWare News.

I would also be interested in the idea mentioned some time ago of publishing separately the things that never made it into SWN such as contest entries, etc. A compilation like SWN Vol. 1 with new stuff would be pretty interesting.

I like the illustrations added to recent issues. They add a nice bit of color to an already classy publication.

Mel Richardson
Windsor, Ontario

Mel, Our Journal may yet someday become a reality, although we're a bit afraid such an undertaking might result in a publication that would be priced too high to generate enough sales to be worth the effort. How do the rest of you feel about this? How much would you be willing to pay for something similar in size to the SWN Volume 1?

Tom Woods has been our artist in the recent issues. He was beginning to think no one noticed or cared. Your comments have caused one problem though. It's awfully tough to face a New Hampshire winter when one can't find a hat large enough! Just kidding
Tom!- ed.

Word on MemoNotes:

Fred Nachbaur has announced that it is highly unlikely that there will be any more formal issues of MemoNotes. He feels that there isn't enough interest at present to justify the time and expense of producing the newsletter. MemoNotes subscribers haven't been left out on a limb, however. Fred invites subscribers with any additional questions to write to him directly for personalized handling of any questions you still may have.

Border Wars

Dear Editor, I am writing in regard to the upper screen 'tearing' problem described in T. J. Morley's letter that appeared in issue 4:5. I have suffered through the identical problem ever since I got my 2068 several years ago. I have tried every adjustment on both the 2068 and the Sears TV/monitor, but the problem is still not corrected. Last week I determinedly tried again. I finally did away with the tearing problem, but was left with a choice of washed out INK colors or an acceptable black INK with the whole spectrum of colors rolling through only the INK portion of the screen. At the present moment, I have made adjustments to allow the latter, but as I type this letter, all the characters on the screen are ALIVE. But they are readable.

My solution for the last several years has been to routinely avoid the use of colors 1, 2, and 3 in my own programs. In the programs that I have purchased, I have managed to track down the code using or assigning colors 1, 2, and 3, and have changed these to colors 4, 5, and 6. If necessary, changing the colors to use only 0 and 7.

I do have two additional observations on this problem that were not mentioned in either Morley's letter or your commentary. One is that the problem seems to be basically of contrast between the chosen BORDER and PAPER colors. Colors 0, 1, 2, and 3 work fine by themselves or contrasted with colors 4, 5, and 6, but colors 0, 1, 2, and 3 cannot be contrasted with each other. Colors 0 and 7 contrasted as PAPER and BORDER start the screen rolling which no adjustment of the monitor will stop.

The second observation is that hooking the computer output to the TV input of the Sears TV/monitor, rather than the monitor input, completely solves the problem--no tearing of the screen at all. Unfortunately, this sacrifices the benefits of monitor use.

Lou Dooley
Ocala, FL

Serial I/O Ports

Dear Editor, I recently wrote to Protecto Enterprises asking if the Big Blue Printer they advertise could be used with a Timex TS2068. Their reply stated that if the computer had an RS232C Serial Port and can be adjusted properly, the printer would work. I could not find an RS232C serial port in the TS2068 Technical Manual.

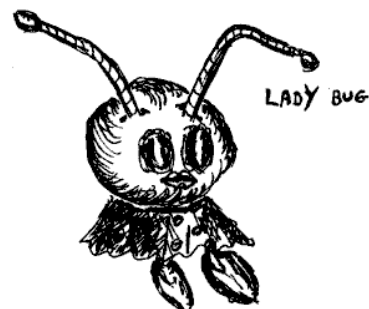
Do you know if this printer can be used with my TS2068?

Earl F. Swallow
Manchester, CT

Earl, The TS2068 does not, as you have discovered, have an RS232C port. This means that you need a "printer interface". Serial interfaces for the TS2068 are supplied by Byte-Back and Aerco. The Sinclair "Interface I" also has a serial port, if memory serves me correctly. Parallel interfaces are supplied by more manufacturers, including Byte-Back, Aerco, and Tasman.

Beside the appropriate interface, software is required to run it. The TS2068 is nice in that it boots its command vector tables into RAM where they can be changed. This means that suitable 'interface driver software' can be loaded into high memory, making the BASIC LPRINT, LLIST, and COPY command direct output to the "big printer" interface instead of the TS2040. Even with this software you might have trouble with programs that are not made to handle a printer other than the TS2040.

My recommendation would be to go with a parallel interface and printer. That approach is usually "plug-n-go". With a serial interface, there are many parameters that must be adjusted and/or met for the thing to work, which can sometimes be extremely difficult to do.-Fred.



Spectrum PRODUCTS

Dear Editor, I currently own a ZX Spectrum (the 'real' one from the U.K.) along with a ZX Interface I and a ZX Microdrive. I am interested in locating a vendor who handles the Microdrive cartridges (I understand they are identical to the type used for the QL) as well as Spectrum hardware and software. Can you make a recommendation?

Petrus Hu
Ames, IA

Petrus, The first name that comes to mind would be Curry Computer, PO Box 5607, Glendale, AZ 85312-5607, (602)978-2902. They carry a complete line of hardware and software for the Spectrum and the QL. You also might like to contact some of our other advertisers and those listed in the Support column.-ed.

SyncWare Examines the IBM PC

PART 2. A Bit About the PC

First, it is important to tell you what we mean by the term "PC". IBM started 5 years ago with their first model, the IBM PC, which meant "Personal Computer". There was an early rush by about a dozen companies to make "PC Compatible" computers. That is, computers which were functionally the same. The intention was to make alternative machines at a lower price which would run the same software as the IBM PC. For the most part, these compatibles were very different from the IBM computer electronically. From the software standpoint, different brands offered differing degrees of compatibility.

In just the last year, manufacturers have been able to reproduce both the electrical AND functional aspects of the IBM exactly. It is virtually impossible to tell the difference. We use the term "PC" to describe all of these EXACT reproductions. Other terms which have equal meaning are "Clone" and "PC Compatible". IBM came out with an improved version of the PC which it called the IBM PC/XT. Most clones on the market offer full compatibility with both IBM models. There are, however two other IBM models which are NOT to be confused with the PC: the IBM PC Jr. and the IBM AT. These machines will run much of the same software as the PC, but electronically, they are quite different.

To one who has used a Timex computer, the PC is a very interesting machine because in some respects, it is the exact opposite of a Timex, yet at the same time, it is startlingly similar. Let me explain.

The Opposites

My overall impression of Timex computers is that it offers the finest Basic language, the most efficient use of memory and the most sensible operating system of any other computer. However, it fails horribly in a crucial aspect that practically ruins its usefulness. Timex hardware is awful. On the TS1000, overheating and rampack wobbles, the poor quality video, the membrane keyboard, and unreliable, slow cassette loading are enough to try the patience of Job. Hobbyists and experimenters have made great attempts to correct these shortcomings, but in most cases, it involves a great deal of tinkering to get

everything working right, and even more tinkering to keep it that way. The TS2068 fixed many problems of the 1000 computer, but the "wavy" video, still slow tape loading, and bad keyboard leave many problems only slightly corrected.

The PC is just the opposite in this respect. It gives you very nice hardware and software that stinks. You get beautiful video: clear, sharp, rock steady, able to be read from opposite sides of a room. You get just a ton of memory. Most clones come with 640K built-in. The power supply gives no crashing problems, and disk drives are fast, reliable, and store gobs of data. Anyone who has fussed and fumed over a Timex will think they've died and gone to heaven when they start working with the PC hardware.

Looking at the operating system and its associated software, a Timex enthusiast would see a real mess however. It appears to be a hodge-podge of routines which were written by a hundred different people. Everything was pieced and patched together, but nobody ever really cleaned it up. As a result, it is very wasteful of memory space, slow considering the speed of the microprocessor, and quite cumbersome. The Basic programming language consumes over 60K. Syntax checking is completely lacking, programs can be a maximum of 64K in length (even if you have 10 times more memory in the computer!), and a faulty program line can crash the PC like a bad USR call does on the Timex.

I find it to be particularly ironic that we ZX/TS fans have put up with continual harassment from PC users who told us we had nothing more than a toy when in actuality, it was they who were forced to work with such inferior software. They didn't know what they were missing, and we didn't know how good we had it!

The Similarities

After all my complaining, you might wonder why we even suggest bothering with the PC. Actually there are a lot of reasons. They revolve around the similarities I see between the PC and the Timex.

When you strip away the bad operating system you are left with a wonderful machine. It is extremely simple the way it works. There is nothing technologically advanced about it. There are no mysterious chips, esoteric functions or fancy hardware. At its gut level, the PC is nothing more than a very souped up ZX81. A machine code programmer can take complete control of the machine and make it do whatever you want. This is in contrast to the more technologically advanced computers like the TS2068, the QL, the Atari, Amiga, and Macintosh where sophisticated hardware can often make a programmer's life miserable.

Also of interest to the machine code programmer, is the 8088 microprocessor in the PC. This chip comes from the same roots as the Z80 which means that the methods and philosophies used with the Z80 can also be applied to the 8088. Many MC instructions are identical or are only semantically different. For example, the Z80 term, "LD A,00" becomes "MOV AL,00" in the PC. It is very easy to make the transition.

For the Non-machine code programmer, who's probably thinking "big deal" at this point, these similarities have important consequences for you too. Because the PC is so easy to program in machine code, many of the shortcomings found in the operating system have been circumvented by

applications software or additional system software. This stuff is widely available in the public domain libraries I mentioned in the last section on the PC. Probably half of all software in these libraries could be categorized as "Utility" software: Toolkits, patches, and enhancements which fill the holes left by the operating system.

A MODULAR Computer

Unlike Timex computers which come, more or less as a complete unit, the PC is a "modular" machine. By this, I mean that different components are assembled into one complete system. This is a very nice feature because you can pick and choose your components so the computer ends up exactly the way you want it. It also means that you can buy a PC practically as a kit and assemble it yourself just by plugging the right boards together. Unfortunately for many who are used to buying a unit they can just plug in and go with (like a Timex computer), this modularity can be a mixed blessing if you don't know which components you need to get in order to have a fully functioning PC. Even for the experienced hacker, it seems that no matter how many components you add, there's always one more that you haven't got. But even Timex machines are like that, it seems.

To avoid disappointment, it is very important to know what parts you need to get BEFORE you place an order. Many vendors sell complete packages often at a significant price savings over buying individual parts yourself. But even with these packages, there is a wide variety. You need to know what is included and what is not. Below I have listed the components necessary to have a functional PC system.

1. Motherboard:

This is the main circuit with the ram and micro processor on it. It also includes a keyboard connection, and a number of expansion slots (usually 8). Some boards come with just one clock speed like the PC. Others called TURBO boards give you two or more speeds. TURBO boards generally boost computer speed about 50%. Motherboards are often sold without ram chips installed. If the ads say "OK RAM", be prepared to spend \$20 more for ram chips. 640K is the most memory the PC is able to handle without additional bank switching hardware. Be wary of motherboards which do not at least provide sockets for this much ram. If they don't, it means that you'll need to buy a ram expansion card.

2. Keyboard:

Detached keyboards have a coiled cable which stretches to the back of the PC and plugs into a connector on the motherboard. Be sure the cable is nice and long. Several styles of keys are available. All are a dramatic improvement over the Timex, but some PC keyboards, never the less, are better than others. I prefer one with large ENTER and SHIFT keys. These keys on some boards are no larger than the regular alpha keys.

3. Power Supply:

Unlike the Timex power supply which is measured in Millamps, the PC supply is measured in WATTS. It includes a fan and ON/OFF switch. You should probably look for the highest wattage you can afford. As you add peripherals, power consumption goes up. If you plan ahead, you won't need to buy a

larger supply later. A good size to start with is something between 100 and 200 watts.

4. Video Adapter:

There is a large and confusing variety of video adapters on the market. You must be careful to choose the one to suit your needs. Some of the types are listed below:

monochrome video adapter: this is the cheapest and requires the use of a TTL level monitor--NOT a TV set or a composite video monitor. This adapter will only print text and block graphics. You cannot plot individual pixels. But what you do get is absolutely beautiful text displays at a reasonable cost.

monochrome graphics adapter: Same as above. Only you can plot pixels on the screen if you can figure out how. The Basic language will not let you access pixel graphics with this adapter. There is very little information out which tells how to work the graphics mode of the MGA. However it will give extremely high resolution (720 X 340), and in text mode, the picture is stunning. This is my personal favorite for word processing. It has two 32K display buffers which eliminate an annoying flicker (you heard it right--screen flicker) which occurs with the other types of display adapters. This board also has a built in parallel printer interface. The first adapter of this type was called the "Hercules Graphics Adapter" which set something of a standard used by virtually all other monochrome graphics boards.

color graphics adapter: This board (the CGA) is perhaps the most frequently used video card. It prints text and also lets you use Basic to access individual screen pixels (640 X 200). Several modes of operation with up to 16 colors are possible. This adapter works with either an RGB or a composite video monitor.

5. Disk Controller:

Some controller cards allow you to connect 4 different disk drives. Others like the one I use are called "Multi-IO Cards" which connect just 2 drives, but also provide you with another printer port, 2 RS232 serial ports, and a battery backed clock calendar. Choose the card that fits your needs.

6. Disk Drives:

At least one disk drive is necessary, but making do with just one can be a bother. You are much better off getting two disk drives. When shopping for drives, don't go for the bargain basement specials unless you are sure they will work reliably. Remember, they probably are on sale for a reason. It may be that they are not up to snuff. If you buy a package deal from a vendor, be sure you know how many disk drives come with the base unit. Often, the second drive costs extra.

7. Case:

The motherboard, power supply, disk drives, and assorted expansion cards are all made to fit inside a big steel enclosure. Cases come in two flavors: the flip-top and the slide-in. I understand that the flip-tops often have trouble passing FCC noise

regulations, but they offer convenience in getting to the various components inside.

8. DOS

All the other components were hardware items. DOS (or MSDOS--the Disk Operating System) is the one software module which is absolutely essential. Without it, your system is worthless. The operating system comes on a disk. When you turn the computer ON, the DOS is loaded into memory and runs. This is the software which reads the disk drives, keyboard, etc. Frequently, vendors who sell package deals do not include DOS with the computer, however many do. Don't expect it unless the dealer specifically states that DOS is included. You may need to buy MSDOS as a separate item.

The DOS disk contains many useful utility programs and the BASIC programming language. BASIC is an interpreter like Timex Basic. There are many different versions of BASIC. Many are quite different in the way they work, but all are functionally about the same.

Finding a Supplier

The list above shows all the parts necessary to put together a complete working system. As with buying a car, there are, of course, many "extras" you can add if your needs and pocketbook allow it. Finding a reliable dealer to supply all these components is not always the easiest proposition, however.

A brief look through a magazine like "The Computer Shopper" will reveal over a hundred different places to look, but how do you know which dealers are honest and reputable? I recommend three methods for finding a good supplier:

1. Rely on product reviews or advice from people you know who have purchased a PC.
2. Buy a name brand product like an Epson or Tandy.

3. Go to a dealer you have dealt with before. Many Timex dealers are now branching out and offering PC compatibles. E. Arthur Brown Company, Knighted Computers, TEJ Computer Products, and HAL-Tronics are some that we know of.

Prices vary dramatically from dealer to dealer. We have seen them range from \$500 to \$1000 depending on what you get.

Continuing the PC Saga

There is still a LOT more to be said about the PC. You can expect to see more in future issues of SWN. While you're waiting for us to get the next issue out, let us refer you to a few other sources of PC information which we have found useful.

Soft Sector Magazine, Dept. SWN, P.O. Box 385,
Prospect, KY 40059
(the SYNC Magazine of the PC world)

1001 News, Rt. 1, Box 112, McHenry, MS 39561
(A nice beginners publication for the Tandy 1000 models)

PC Resource, 80 Elm St., Peterborough, NH 03458
(Full of Ads and a lot of good nuts and bolts info)

MicroCornucopia, P.O. Box 223, Bend, OR 97709
(The SWN of the PC World... At least for now!)

E. Arthur Brown Co., 3404 Pawnee Dr., Alexandria, MN 56308
(sells Tandy computers at discount)

Hal-Tronix, P.O. Box 1101, Southgate, MI 48195
(sells generic PC clones)

Thomas B. Woods, Box 64, Jefferson, NH 03583
(that's me! sells PC Findex aka Pro/File)

TYD★BYTS

Stuff from the desk of Kent Cook

OLIGER PRINTER PORT/MSCRIPT COMPATABILITY

If you have a slight amount of hardware expertise, here is a simple, no-extra-parts-required, modification for the JLO parallel port which provides support for the Centronics port PAPER EMPTY status bit. This also eliminates the diode-ORED pull-down, which is, theoretically, a marginal technique (although it's cheap and seems to work fine!). I don't know if this is complete AERCO compatibility. It does provide all the status signals required by MSCRIPT's AERCO compatible software.

You must be able to open the Centronics connector on the printer cable. If your's is a solder-on type, you're in luck. IDC connectors can be managed, but they are more difficult. If you have a cable with the connectors molded on you may have to cut off the Centronics end and replace it with a solder-on type. The JLO port docs. give the pin connections. Remove the wire from pin 10, attach it to pin 12 and put the connector back together. Pin 10 is the active low ACK(nowledge) signal from the printer. It was

not used in the JLO port, so we didn't lose anything.

Remove U2, the 74LS126 IC, from its socket and solder a jumper between U2 socket pins 2 and 3. This frees a section of the 'LS126 for our use. Bend pins 1, 2, and 3 of the 'LS126 straight out from the IC and re-install it so the three pins do NOT enter the socket.

Now install three jumper wires between the exposed IC leads and the circuit board. Insulated 28 or 30 gauge wrapping wire is ideal. Run a jumper around the side of the circuit board from U2 socket pin 4 on the bottom of the board to pin 1 of the 'LS126. Next, jumper pin 2 of the 'LS126 to pin 10 of the header connector (the pin closest to the 'LS02). Finally, run a jumper around the side of the board from pin 3 of the 'LS126 to socket pin 17 of the 'LS374 (where the diode anode is connected in the modification shown in the JLO port docs).

That's all!

2K-SDP:

A Serial Data Port

A system which allows transfer of data between ZX/TS and TS2068 computers and cassette tape operations in TS2068 format.

by Kent E. Cook

The PROBLEM

Have you ever needed to move memory contents from one type of machine to another? I have a 2764/27128 EPROM burner on my TS2068 system and there have been several occasions when I've wanted to use it on code which I wrote on and for my ZX81/TS1000 system. Having disk storage, an RS232 port, modem, or full-sized printer on only one system are some other motives.

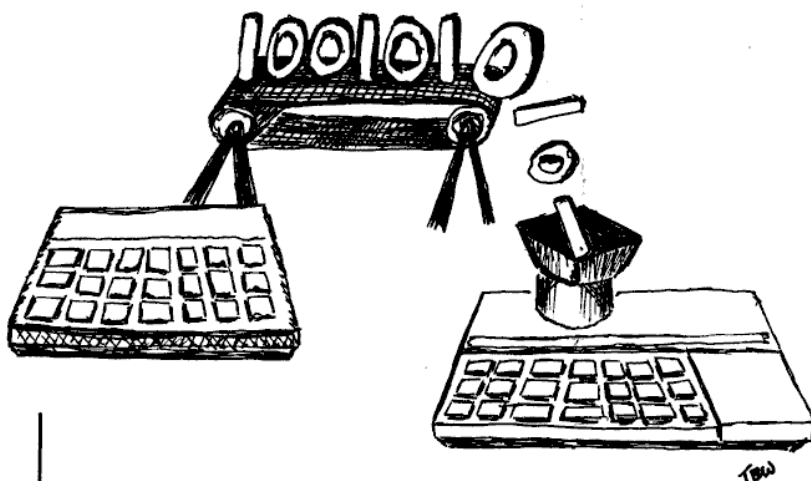
The SOLUTIONS

My eventual solution to this will be a 32K block of NV-RAM which can be addressed by either machine. (Have you noticed the 32K X 8 bit static(!) 150 ns RAM chips advertised for less than \$15? My, how times change!) This, however, being a hardware solution, requires time and money.

The next best solution is parallel transfer between machines. Ideally, you would use two bi-directional ports, such as Tom Woods' Universal I/O. An existing printer port could be used for the transmitter but you still need one input port to receive, and would have to re-configure the system to change direction. To circumvent the lack of handshaking, the data could be sent in 4-bit nibbles using the other bits to signal start of data, upper/lower nibble, end of byte, end of data, etc. Once again, this is a hardware solution and requires money.

A third possibility is serial transfer. If you have an RS232 port or a modem on both systems, you're in business. Again, hardware is usually not free.

ZX81/TS1000 and TS2068 computers have built-in serial ports and a software solution to data transfer is possible using Firstloadr or similar commercially available products. Put the code in REM in the ZX/TS system and save it directly into the tape input port on the TS2068 while executing the loader program. A more elegant method would be using HotZ II on the ZX/TS to feed the LD81 routine contained in early versions of HotZ 2068. These methods use the least hardware, (an amplifier on the ZX/TS output) and are the cheap. The compromise is in speed of transfer and accuracy.



The SYSTEM

I have always thought the programs which permitted the TS2068 to load ZX/TS tapes were examples of the tail wagging the dog! The TS2068 tape system is obviously superior, so why not implement it on the ZX81/TS1000? In keeping with the Sinclair tradition of using a minimum of hardware with ingenious software, this idea provides a more versatile tape storage facility on the ZX/TS and a cheap method of data transfer between different systems.

Unfortunately, the ZX81/TS1000 will not directly emulate the TS2068 cassette tape operations because of minor hardware problems. These are a direct result of the different software approaches to serial data transfer. The ZX/TS save routine generates a string of fixed frequency pulses to represent the serial data. The number of pulses between a blank period determines a space or a mark. The TS2068 transmits one frequency modulated cycle per data bit with the period of the cycle set by the logic level of the bit. The hardware incompatibility results chiefly from the ZX/TS input/output conditioning. The RC voltage-dividers and filters pass the ZX/TS signal spikes but destroy a signal of the type required for the TS2068 software.

The HARDWARE

I had some goals in mind when I started to develop a hardware interface between machines. It could not require modification of both machines, a separate power supply, or interfere with the normal use of the tape ports. It had to use inexpensive, common components, and it had to work. After considerable tinkering, I arrived at a circuit which uses one N-channel enhancement mode MOSFET transistor, one CMOS hex inverter IC, and a dozen passive components. It cost should be less than \$10.

My system shed its case some years ago, so component placement within the case is academic for me. I'd bet that most fellow-tinkerers have the same situation. However, if you are using a ZX/TS in the original case, you might be able to get the entire circuit under the hood on a piece of perfboard. Wire it with 28/30 gauge wrapping wire, and bring the input and output to subminiature (3/32 in) jacks mounted in the case.

An alternative method would be placing only the resistive voltage divider which is attached to the logic chip tv/tape output and the transistor with its gate resistor/capacitor inside the case. Connect the divider output and the transistor input to jacks. This provides isolation for the logic chip pins. I strongly urge that you do NOT bring the logic chip pins directly to the outside world!

The easiest place to pick up the tv/tape output is at the modulator. The tape input is pin 20 at the corner of the logic chip. Connect to it at the input series resistor (a 4.7K adjacent to pin 20 on my old ZX81 board).

The IC is a 74HC04. Although a 74C04 worked in my system, the 'HC04 has "improved" characteristics. A 4069 CMOS hex inverter might work and the pinout is the same - I did not try any 4000 series devices. The transistor is a Radio Shack 276-2074. The caps are ceramic disks and the resistors can be 1/4 or 1/8 Watt carbon or molded composition types. Nothing is critical. However, the caps and some of the resistors are parts of time dependent networks and the values and/or ratios can not be changed too much.

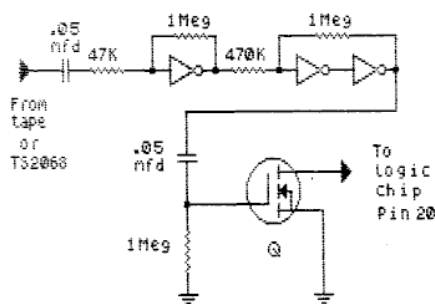


Fig. 1 Input

The input portion of the interface is depicted in Fig. 1. The signal from a TS2068 tape output ("mic") port or a tape recorder earphone jack is AC coupled to an inverter biased into the linear region with negative feedback as an amplifier with a theoretical gain of approximately -20. On paper, a .25VAC PP signal will cause the output to swing the full 5V of the supply.

The first stage output is coupled to a pair of cascaded inverters connected with positive feedback as a Schmitt trigger circuit. The ratio of the input resistor and feedback resistor provides a hysteresis of 2.5V. This provides "snap action" and eliminates the garbage caused by the first inverter being biased at the trip point.

The output of the 3rd 'HC04 is AC coupled to the N-channel switch. In the absence of an AC signal, the transistor drain floats, having no effect on the operation of the normal tape input to the logic chip.

The output circuit, shown in Fig. 2, is even simpler.

The ZX/TS logic chip output has voltage excursions from 4V to 2.4V. This is reasonably close to the logic levels of a CMOS device operated from a 5V supply. In fact, I found that 74C04 and 74HC04 devices would recognise the logic chip tv/tape output pin levels directly. The low level is too

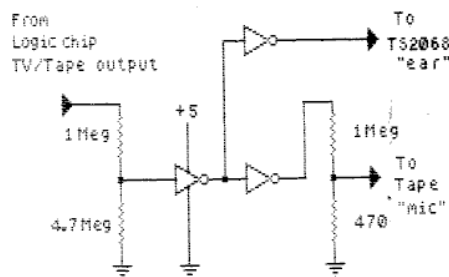


Fig. 2 Output

close to the midpoint to be comfortable, so it is fed to a resistive voltage divider which reduces the excursions at the input of the 'HC04 inverter to 3.2V and 2V. The output of the 1st inverter is connected to two others. One provides a 5V level output for the tape input ("ear") port of a TS2068 for data transfer between machines. The other inverter has a divider to provide a signal for a tape recorder microphone input.

The input and output will interact as they do in a normal ZX/TS save/load. Disconnect the input when making a save. The high impedance of the input amplifier makes it sensitive to hum and tape hiss. Don't start a "2K-SDP" load or verify until the leader tone is present.

You MIGHT get surprising results by using this port hardware with other cassette software. I use it with ZXLR-8 and it actually loaded an old intractable "Packrabbit" tape with the IPS "FASTLOAD" software.

Another benefit I realized was being able to use a HiFi cassette deck which did not have enough output at the earphone jack to drive the ZX/TS load routine.

The SOFTWARE

"2K-SDP" is 1.5K bytes long. This version is mapped at 7A00 Hex, assuming it will be used above RAMTOP. This results in a calling address of 31232 Decimal. The screen prompts require an "expanded" display file. The program requires 38 bytes in the E_LINE area of RAM for headers, flags, etc. It will report ERROR 4 if there is not enough space below ramtop.

The tape version of the program, available from me, moves RAMTOP, and provides a means of making a back-up copy. The best place for this (and any other utility or operating system) is EPROM. It fits a 2716 with half a "K" left over and could be used in a Hunter Board or similar EPROM reader.

The system has consistently generated error-free tapes and data block transfers of more than 32K bytes. It is processor clock dependent, and will require modification to work with a system not running at the standard rate. Modification may be made by simply changing the values in the software timing loops based on the difference in system clock rates.

The routine is entered from BASIC with "RAND USR 31232". The display prompts for SAVE, LOAD, or VERIFY. Hitting "Q" from this display returns to the caller. Although the code does the prompts and inputs in SLOW for flicker-free display and the save/load/verify in FAST, it returns to the caller

The software then prompts for a tape name of 0 to 10 characters, accepting numbers and letters only; no punctuation, spaces, or tokens. Short names are terminated with "ENTER" and the "DELETE" key removes the last character entered. The name is translated to ASCII so it can be read by the TS2068 and restricting it made the code simpler. The save routine must have a name entered. Load and verify may be executed without a name by hitting "ENTER" and will then operate on the first data encountered. Entering a name for load and verify makes them selective, just as the ROM routines do.

A data block save then prompts for starting address and block length. Both may be entered in hex or decimal. Hex entries must be prefixed with "H" and must be 5 characters including the prefix. (I.E., H0300, H7FFF, H0000, etc.). Decimal numbers may be 1 to 5 characters. Decimal entry may be terminated for less than 5 characters with "ENTER". The "DELETE" key works as it did in name entry. The saves may be made from any address and have any length except a block length of zero or an address and length which total more than 65535 Decimal. Incorrect parameters take you back to the address prompt.

There is a short blank period before and after the save to allow time to switch the recorder or start the TS2068 routine for transfers. The save may be stopped with "BREAK" which returns to the first screen. A completed save returns to the caller.

A program load/verify specifies a starting address of 16393, the same as the ZX81/TS1000 routines. It also specifies a length of zero which may seem a bit odd. This is the signal for the loading routine to use the length found in reading the header.

1 = CODE	Address and length from incoming data
2 = CODE MMMM	Specified address
3 = CODE MMMM,NNNN	Specified address and length

[illegible]


```

29 LD H,29
ADD HL,HL
SCF
LD HL,(3838)
NOP
INC D
NOP
RST 38H
$BLN ADD HL,HL
LD H,39
LD H,00
LD SP,332A
INC L
ADD HL,SP
DEC L
NOP
NOP
INC D
NOP
RST 38H
$COD JR Z,7FA7
ADD HL,HL
LD HL,(00FF)
LD (3232),A
LD (1AFF),A
NOP
INC SP
INC SP
INC SP
INC SP
RST 38H
$Y/N INC (HL)
JR NC,7F86
DJNZ,7FC6
JR 7FBD
LD DE,FF0F
$PRE LD L,33
JR Z,7FC6
SCF
SCF
LD HL,(3928)
NOP
DEC (HL)
LD H,37
LD H,32
LD HL,(2A39)
NOP
$SPC LD HL,(2E28)
LD HL,2A
LD L,2A
ADD HL,HL
LD C,FF
$TAP ADD HL,SP
LD H,35
LD HL,(000E)
RST 38H
$CSE JR Z,7FE2
LD HL,(3028)
LD C,7FF4
LD (2A00),A
NOP
SCF
INC (HL)
SCF
LD A,(DE)
NOP
DEC SP
LD HL,(2E37)
DEC HL
LD A,00
ADD HL,HL
LD H,39
LD H,FF
$URE DEC SP
LD HL,(2E37)
DEC HL
LD L,28
LD H,39
LD L,34
INC SP
NOP
LD HL,(3737)
INC (HL)
SCF
NOP
LD H,39
NOP
RST 38H
$CON DEC (HL)
SCF
LD HL,(3838)
NOP
LD H,33
LD A,00
JR NC,RFCN
LD A,00
ADD HL,SP
INC (HL)
NOP
JR Z,802D
INC SP
ADD HL,SP
LD L,33
LD A,(FF2A)

```

The parameters and the final load/verify prompt are now displayed. A negative response returns to the first screen.

Because zero is used to denote an unspecified address (or length), data can not be loaded or verified starting at address 0, rarely a hardship for most systems. (Remember, data can be saved from 0 to copy ROM code to an EPROM burner driver, for instance, and will, therefore, reload to that address).

When the header is loaded, the specified parameters are compared with those in the header. Out-of-range parameters are rejected and both sets are displayed for examination. The program will reject a specified length greater than the length on the header, and will reject an address and length which total more than 65535.

A successful load or verify simply returns to the caller. Any load resulting in checksum failure terminates with a screen which displays "CHECKSUM ERROR". A bad verify goes to a screen which displays the address at which the verification failed!!

"2K-SDP" will load and verify data blocks which are shorter than the original save. The checksum contained in the save is, therefore, not available and the load routine exits to the "CHECKSUM ERROR" screen.

"BREAK" was not implemented on the loading loop to make timing easier. To abort a load or verify just stop the input. To exit from the first part of the loading routine (which looks for a leader), stop the input and press "BREAK".

To make the loading software less machine dependent, I wrote it to calculate its timing based on the leader frequency. The negative side of the coin is the software reacts to hum, tape noise, and tones which are not a leader. To avoid problems, don't answer the final load/verify prompt until the leader signal is present at the input.

Loading this much code by hand can be a disaster. I've included a HOT-Z 2068 disassembly (Yes, 2068 - I don't currently have a TS2040 on my ZX/TS system, so I moved the code and name file to my TS2068 with "2K-SDP") and a hexadecimal table. The last 256 bytes are print strings, not machine code. The table is in 220 byte blocks. It starts at 16514 and reads left to right in rows from top to bottom. Good luck!

Software Instructions

This is a long MC program to load by hand. Mr. Cook, the author, is offering to sell a tape of the software for \$10. If you are a beginner, we would highly recommend purchasing the tape. Mr. Cook can be contacted at:

Mr. Kent E. Cook
P. O. Box 1031
Terre Haute, IN 47808-1031

If you should decide to do-it-yourself, please follow the procedure given below.

STEP 1:
Create a 0 REM line at least 1564 bytes long using either a REM Generator utility or the procedure shown in step #2.

STEP 2:

If you have used a REM Generator utility to create a 1564 byte long 0 REM line, proceed to Step #3.

A: Enter the following line exactly as shown:

```
1 PRINT 0+0+0+0+0+0+0+0+0+0
```

B: Using the Shift-1 edit key, duplicate the line you just entered 16 times. The result should be lines 1 thru 17, all identical.

C: Without a line number, enter the command...

```
PRINT PEEK 18089,PEEK 18090
```

a pair of 118's should appear on the screen. If not go back to step A.

D: Enter the following line:

```
20 REM ***NEVER DELETE OR LIST
LINES 1-17***
```

E: Enter the following commands without line numbers:

```
POKE 16510,0
POKE 16511,41
POKE 16512,6
POKE 16513,234
```

What you have just done is to create a 1575 byte long 0 REM statement. For an explanation of how this works, see "VDAQ", page 63, in Volume 1 of SyncWare News.

LISTING 1: HEX LOADER

```
100 CLS
110 PRINT "ADDRESS=";
120 INPUTT AD
130 CLS
140 PRINT "ADDRESS=";AD
150 LET SUM=0
160 LET A=AD
170 FOR L=1 TO 20
180 FOR C=1 TO 30 STEP 3
190 INPUT H$
200 IF H$="" THEN LET H$="00"
210 IF H$="S" THEN STOP
220 IF LEN H$<>2 THEN GOTO 190
230 LET N=16*CODE H$(1)+CODE H$
(2)-476
240 IF N>255 OR N<0 THEN GOTO 190
250 PRINT AT L,C;H$
260 POKE A,N
270 LET SUM=SUM+N
280 LET A=A+1
290 IF A=18090 THEN GOTO 320
300 NEXT C
310 NEXT L
320 PRINT AT 21,0;"CHECKSUM=";S
UM;" OK? Y/N"
330 PAUSE 4E4
340 IF INKEY$="Y" THEN GOTO 370
350 IF INKEY$="N" THEN GOTO 130
360 GOTO 330
370 CLS
380 PRINT "SAVE? Y/N"
390 PAUSE 4E4
400 IF INKEY$="Y" THEN SAVE "2K
-SDP"
410 IF INKEY$<>"N" THEN GOTO 37
0
420 GOTO 100
```

STEP 3:

Enter the BASIC program shown in LISTING 1 and run it. This takes data from the HEX TABLES (Table 1) and inputs it into the "mega-rem" you just created. Enter the hex code data from left to right and top to bottom. Data entry may be halted for the tables that are not a full 220 bytes long by simply entering the letter "S" at the data input prompt.

STEP 4:

Once all the data has been entered, delete all the program lines greater than 20. This will leave the 0 REM statement and line 20. Enter the BASIC program shown in LISTING 2. Save your program by typing RUN. This will create a program that will self start upon loading. The screen prompts and this article contain the rest of the information you need to run this program.

LISTING 2: 2K-SDP SOFTWARE

```
100 SAVE "2K-SDP"
110 RAND USR 16514
120 CLS
130 PRINT "2K-SDP NOW LOADED"
140 PRINT
150 PRINT "RAND USR 31232 TO RU
N"
160 PRINT
170 PRINT "PRESS ""S"" TO SAVE
2K-SDP"
180 "ANY OTHER TO RETURN TO BAS
IC"
190 PAUSE 4E4
200 IF INKEY$="S" THEN RUN
210 NEW
```

LARKEN ELECTRONICS**DISK INTERFACES****----- LARKEN 2068 / Spectrum DISK SYSTEM -----**

- The system consists of the LKDOS cartridge and a Double Density rear disk interface. 800K on a Quad Drive
- Fully 2068 / Spectrum compatible / OS-64 compatible
- It uses all standard (Token) keyboard cassette commands CAT MERGE ERASE FORMAT LOAD SAVE PRINT and more
- Uses NO RAM space . HAS 8K ROM and 8K RAM on board
- NMI Memory Save Feature :PUSH-BUTTON program transfer
- A KEMPSTON compatible Joystick port is also on the IF.
- Also, 10 Extended Basic Commands for Graphics,Utilities and up to 3 scrolling Windows on the screen . An Aerco compatible printer driver is also in the Lkdos Cartridge
- The disk interface is a compact rear mounted board that can support 1 to 4; 3" 3.5" or 5.25" SS,DS or Quad Drives
- Easy to setup . 90 day guarantee

***** ATTENTION *** AERCO FD68 and RAMEX DISK USERS**

The LKDOS cartridge is now available for your disk IF's . It will allow your disk systems to be fully Spectrum and OS-64 compatible and Larken disk compatible and have all the commands mentioned above . Also a SNAP-SHOT save button can added . RAMEX users will now be able to use all the memory . AERCO users can now have all of the above features plus the Lkdos uses the Aerco Ram as a RAM-DISK !

PRICES : (US)	2068/Spectrum Disk System	\$119.95
Add \$5 S&H	LKDOS Cartridge (Aerco,Ramex)	\$65.00
	ZX-81 Disk Controller	\$99.00
	256K Non-Volatile Ram Disk	(TBA)
	Drive Floppy cable	\$8.00

* LARKEN ELECTRONICS RR#2 NAVAN ONTARIO CANADA K4B-1H9 *

START ADDRESS=16514
 21 00 7A 22 04 40 2B 36 3E 2B F9
 2B 2B 22 02 40 21 9E 40 11 00 7A
 01 00 06 ED 80 C9 CD A3 14 01 26
 00 CD C8 0E EB CD A6 14 2B 3A 3B
 40 77 2B 2B EB 21 00 00 39 EB 72
 2B 73 2A 14 40 11 22 00 19 5E 23
 56 EB F9 CD 2B 0F CD 17 7E CD D0
 7E D6 36 20 13 CD 2A 0A 2A 1A 40
 2B 7E CD A3 14 E6 40 CA 23 0F C3
 2B 0F D6 02 28 0A D6 F9 28 05 D6
 0A 20 DA 3C 3C 2A 1A 40 2B 2B 77
 CD E9 7A 20 05 CD C9 7D 28 F6 CD
 30 7E CD D0 7E FE 35 20 1A CD C9
 7D 20 0D 01 09 40 2A 14 40 ED 42
 44 4D CD 80 7B 01 09 40 CD 6E 7B
 18 23 FE 29 20 DB CD C9 7D 28 17
 CD F2 7D CD D0 7E FE 1D 28 10 FE
 1E 20 05 CD 63 7B 18 07 FE 1F 20
 EC CD 56 7B CD D3 7D 11 83 7F CD
 54 7E CD D0 7E FE 76 28 05 FE 3E
 CHECKSUM=22255

START ADDRESS=16734
 C2 2C 7A CD 23 0F 21 7A 40 06 1E
 36 00 2B 10 FB CD E5 08 CD 2A 0A
 CD C9 7D 20 08 CD DF 7B CD 8A 7C
 18 03 CD 97 7C C3 36 7A 2A 14 40
 36 03 23 E5 06 0A 36 20 23 10 FB
 06 17 36 00 23 10 FB 11 28 7F B7
 28 09 3D 11 2E 7F 28 03 11 34 7F
 CD 4C 7E 60 69 36 0B 23 E5 06 0A
 36 0B CD D0 7E 57 3E 0A 88 7A 28
 0A FE 77 20 06 36 00 2B 04 18 EA
 FE 76 28 08 FE 40 30 E4 77 23 10
 DE 36 0B D1 E1 3E 0A 90 47 20 03
 36 FF C9 1A C6 14 FE 3A 38 02 C6
 07 77 23 13 10 F2 C9 CD 63 7B C5
 CD 73 7B 0B E1 09 38 F4 C9 11 51
 7F CD 50 7E CD 8B 7B 38 F5 11 0D
 00 18 10 11 61 7F 06 05 CD 56 7E
 CD 8B 7B 38 F3 11 0B 00 2A 14 40
 19 71 23 70 C9 60 69 22 16 40 06
 05 36 0F CD D0 7E 57 3E 05 B8 7A
 CHECKSUM=20334

START ADDRESS=16954
 28 12 FE 76 20 04 CB 5B 28 19 FE
 77 20 0E 36 00 2B 04 18 E2 1E 2C
 FE 2D 28 05 1E 26 BB 30 D9 77 23
 10 D3 36 00 CB 5B 20 08 DF CD 48
 15 CD 8A 15 C9 CD D1 7B 47 E7 D6
 1C 17 17 17 17 4F E7 D6 1C 81 4F
 C9 CD 8A 7C 21 00 10 CD 43 7C 01
 00 2F 3E 00 CD 65 7C 2A 14 40 11
 11 00 06 2A CD 18 7C CD 8D 7C 21
 80 07 CD 43 7C CD BC 7D C5 CD B7
 7D 50 59 E1 3E FF 01 00 13 CD 65
 7C 06 2D 7E CD 65 7C 06 2B 23 1B
 7B B2 20 F4 04 79 CD 65 7C 06 34
 10 FE F5 C6 00 D3 FF 7E 97 F1 C9
 F5 3E 7F DB FE 1F 30 02 F1 C9 C3
 1D 7A 06 93 10 FE CD 36 7C 06 92
 10 FE CD 2D 7C 2B 7D B4 20 ED 06
 27 10 FE CD 36 7C 06 2E 10 FE CD
 2D 7C C9 F5 A9 4F F1 37 CB 17 C8
 10 FE 30 04 06 3C 10 FE CD 36 7C
 CHECKSUM=23610

START ADDRESS=17174
 06 36 10 FE 30 04 06 3C 10 FE CD
 2D 7C A7 06 34 18 E0 CD 8D 7C 06
 03 2B 7D B4 20 FB 10 F9 C9 97 D9
 47 4F 5F D9 57 06 9E 62 6A CD 46
 0F 30 99 CD 89 7D 28 F4 5F 19 10
 F2 7C 2F D9 57 D9 CD 5B 7D FE 00
 20 DB 2A 14 40 11 11 00 E5 D5 19
 CD 50 7D CD 7A 7D D1 E1 20 E6 23
 3E FF BE 28 0C EB 19 06 0A 1A BE
 20 DB 23 13 10 F8 CD B2 7D C5 CD
 BC 7D E1 79 80 28 02 69 60 E5 CD
 AD 7D C5 CD B7 7D E1 79 80 28 06
 ED 42 38 12 69 60 D1 EB E5 2B 19
 E1 38 08 CD C9 7D 3D 28 07 18 18
 CD 87 7E 18 36 CD 5B 7D FE FF 20
 97 CD 50 7D CD 7A 7D C8 CD AE 7E
 18 23 CD 5B 7D FE FF 20 84 06 01
 3E 10 CD 8B 7D CB 10 30 F7 79 A8
 4F 78 BE 20 07 23 1B 7B B2 20 E8
 C9 CD B6 7E C3 1D 7A CD 7A 7D 70
 CHECKSUM=26207

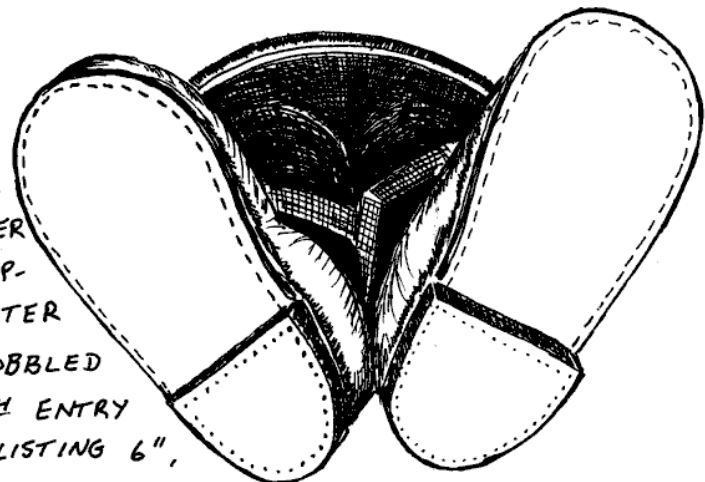
START ADDRESS=17394
 23 1B 7B B2 20 F6 C9 D9 06 00 D9
 CD 46 0F 30 E9 06 00 CD 89 7D 28
 F4 30 F7 10 F7 D9 06 FF D9 CD 89
 7D 38 FB 0E 00 06 01 3E 10 CD 8B
 7D CB 10 30 F7 79 A8 4F C9 3E 28
 3D 20 FD D9 4B 0C 28 14 DB FE 17
 30 F8 4B 0C 28 0B DB FE 17 38 F8
 79 82 D9 D3 FF C9 97 B0 D9 C8 18
 A0 01 1C 00 18 0D 01 1E 00 18 08
 01 0B 00 18 03 01 0D 00 E5 2A 14
 40 09 4E 23 46 E1 C9 E5 2A 1A 40
 2B 2B 7E A7 E1 C9 CD DE 7D 3E 1A
 D7 CD B7 7D 18 0E 11 71 7F CD 50
 7E 03 ED 43 0E 40 CD BC 7D CD 20
 15 C3 DB 15 11 1F 7F CD 50 7E 3E
 1D CD 12 7E 3E 1E CD 0B 7E 3E 1F
 CD 0B 7E 60 69 18 66 CD 12 7E 60
 69 18 5F 11 71 7F 18 29 06 00 11
 00 7F CD 56 7E CD 4C 7E 3E 38 CD
 40 7E 3E 31 CD 40 7E 3E 3B 18 10
 CHECKSUM=22251

START ADDRESS=17614
 11 1F 7F CD 50 7E 3E 35 11 3C 7F
 CD 40 7E 3E 29 77 23 97 77 23 36
 14 23 77 23 18 25 06 02 18 06 06
 04 18 02 06 06 0E 00 D5 C5 CD F5
 08 C1 D1 3E 18 90 4F 97 06 20 77
 23 10 FC 0D 23 20 F6 2A 0E 40 1A
 13 FE 40 30 04 77 23 18 F6 E5 3E
 76 36 00 23 BE 20 FA 23 C1 C9 11
 8D 7F CD 4C 7E CD E1 7D CD D6 7D
 11 AC 7F CD 54 7E ED 43 0E 40 CD
 B2 7D CD EC 7D 3E 1A D7 CD AD 7D
 CD EC 7D 18 17 11 B3 7F CD 50 7E
 18 0F E5 11 CF 7F CD 50 7E ED 43
 0E 40 C1 CD EC 7D 11 E6 7F 06 08
 CD 56 7E CD 2B 0F D9 CD BB 02 2C
 20 FA CD BB 02 44 4D 2C 28 F8 CD
 BD 07 7E 21 D2 10 D9 FE E3 CA 1D
 7A FE 77 C8 FE 76 C8 CD D2 14 30
 D9 C9 FF FF FF FF FF FF FF 39
 38 1E 1C 22 24 00 2B 34 37 32 26
 CHECKSUM=24740

START ADDRESS=17834
 39 00 38 2A 37 2E 26 31 00 29 26
 39 26 00 35 34 37 39 FF 38 2A 31
 2A 28 39 0E 00 FF 38 26 3B 2A 00
 FF 31 34 26 29 00 FF 3B 2A 37 2E
 2B 3E 00 FF 35 37 34 2C 37 26 32
 00 FF 29 26 39 26 00 27 31 34 28
 30 00 FF 29 26 39 26 00 26 29 29
 37 2A 38 38 00 14 00 FF 29 26 39
 26 00 31 2A 33 2C 39 2D 00 00 14
 00 FF 28 34 29 2A FF 00 32 32 32
 32 FF 1A 00 33 33 33 33 FF 34 30
 00 10 3E 18 33 11 0F FF 2E 33 28
 34 37 37 2A 28 39 00 35 26 37 26
 32 2A 39 2A 37 FF 38 35 2A 28 2E
 2B 2E 2A 29 0E FF 39 26 35 2A 0E
 00 FF 28 2D 2A 28 30 38 3A 32 00
 2A 37 37 34 37 1A 00 3B 2A 37 2E
 2B 3E 00 29 26 39 26 FF 3B 2A 37
 2E 2B 2E 28 26 39 2E 34 33 00 2A
 37 37 34 37 00 26 39 00 FF 35 37
 CHECKSUM=12627

START ADDRESS=18054
 2A 38 38 00 26 33 3E 00 30 2A 3E
 00 39 34 00 28 34 33 39 2E 33 3A
 2A FF
 CHECKSUM=1218

THIS RARE VIEW
 OF A TIMEX HACKER
 AT 2 AM WAS CAP-
 TURED SHORTLY AFTER
 THE RAM-PAK WOBBLED
 ON THE 2046 TH ENTRY
 OF "LISTING 6".



TS1000 Mystery Program II

by Gregory C. Harder

Here is where we let the cat out of the bag. Load back the program you saved to tape last issue, and enter the following BASIC lines. You will end up with a completely self-documenting arcade-style game program. For those of you who figured out how the game works, congratulations! For those of you who didn't, don't feel too bad. You were good sports for putting up with SWN's warped sense of humour.

In case our printer decides to darken the listing, here are the inverse graphics and letters...

```

12 MAJOR ENTRY POINTS
19 START
30 TITLE
130 PRINT TITLE
270 FROM HARDER SOFTWARE
320 INSTRUCTIONS
350 GAME FEATURES
400 **GRAPHICS**
420 THE MAVZOTH WARRIORS.
430 V<A>
500 THEY ARE SHIELDED BY A FORCE
    FIELD AND CAN'T BE DESTROYED
530 THEIR HEAT RAYS.
600 THE SPIRAL.
650 THE DISINTERGRATOR BEAM.
700 PRESS A KEY TO CONTINUE.
730 THE DEFENDER,graphic A (x9),inverse +
740 ANTIMATTER DEBRIS,graphic A (x4),graphic Y
750 THE PIXELS,graphic A (x2),graphic 1,graphic A
    (x2),graphic 2,graphic A (x2),graphic 3,graphic
    A (x2),graphic 4
760 PRESS Z TO FIRE LEFT
    PRESS N TO FIRE RIGHT
    PRESS X TO FIRE DOWN
    PRESS B TO FIRE UP
800 ENTER GAME SPEED, 1 TO 9
    1 IS FASTEST(SUICIDAL)
    9 IS SLOWEST(BEGINNER)
830 SPEED
880 PRESS A KEY WHEN READY TO PLAY
910 GAME
1000 SCORE
1060 GAME OVER
    FINAL SCORE WAS
    FINAL SPEED WAS
    WANT TO PLAY AGAIN?(Y/N)
1090 END
9994-9999 CONTAIN COPYRIGHT NOTICE AS FOLLOWS
    ARENA*2
    BY
    G.C. HARDER
    SEPT 1984

```

How about a "Mystery Program" for the 2068? If you feel that you have something sufficiently bizarre (and cryptic), send it on!

?

```

5 SLOW
12 REM MAJOR ENTRY POINTS
13 LET SCREEN=16518
14 LET GRAPHICS=16563
15 LET FLASH=18061
16 LET START=17000
17 LET END=16917
19 REM B-E-T
20 LET DF=PEEK 16396+256*PEEK
16397
25 RAND USR SCREEN
30 REM B-E-T
40 DIM C$(8,27)
50 LET C$(1)="
60 LET C$(2)="
70 LET C$(3)="
80 LET C$(4)="
90 LET C$(5)="
100 LET C$(6)="
110 LET C$(7)="
120 LET C$(8)="
130 REM PRINT TITLE
140 FOR N=1 TO 8
150 FOR X=1 TO 27
160 POKE DF+2+X+(7+N)*33,CODE C
    $(N,X)
170 NEXT X
180 NEXT N
190 RAND USR FLASH
200 FOR N=1 TO 20
210 NEXT N
220 RAND USR SCREEN
230 RAND USR GRAPHICS
240 FOR N=1 TO 20
250 NEXT N
260 RAND USR SCREEN
270 PRINT AT 10,5;"FROM HARDER"
280 PRINT AT 23,30;"
290 FOR N=1 TO 40
300 NEXT N
310 CLS
320 PRINT AT 0,9;"INSTRUCTIONS"
    "YOU ARE AT THE CENTER OF AN"
    "ALIEN ARENA AND MUST DEFEND YO"
    "URSELF FROM THE RELENTLESS ATTAC"
    "KS OF THE MAVZOTH WARRIORS, WHO A"
    "RECONSTANTLY FIRING DEADLY HEAT"
    "RAYS IN YOUR DIRECTION."
    "TH"
    "EIR ULTIMATE WEAPON, HOWEVER, IS"
    "THE AWESOME SPIRAL OF OMBIRA WH"
    "ICH CONSUMES ALMOST ALL OBSTI CA"
    "LS IN IT'S PATH."
    "F"
    "ORTUNATELY YOU POSSESS A WEAPON WITH WHICH"
    "TO DEFEND YOURSELF. A FEBRIFUG"
    "E DISINTERGRATOR WHICH DISPERSES"
    "THE HEAT RAYS AND CAN DESTROY TH"
    "E SPIRAL."
    "AT 21,6,"PRESS A KEY"
    "TO CONTINUE."
330 GOTO 330+NOT INKEY$=""
340 CLS
350 PRINT AT 0,6;"GAME FEATURES"
    "1. START WITH TEN LIVES."
    "2. BONUS LIFE FOR EVERY 2000"
    "POINTS SCORED, HOWEVER, TH"
    "ERE IS A MAXIMUM OF TEN LIVES"
    "AT ONE TIME."
    "3. GAME SPE"
    "ED ALSO INCREASES AT EVERY 20"
    "00 POINTS."
    "4. SCORE TEN POIN"
    "TS FOR ALL RAYS AND SPIRALS DE"
    "STROYED."
    "AT 21,4,"PRESS A KEY TO"
    "CONTINUE."
360 GOTO 360+NOT INKEY$=""
370 CLS
380 RAND USR SCREEN
390 FOR N=1 TO 6
400 PRINT AT 0,10;"**GRAPHICS**"
    "AT 0,10;"
410 NEXT N
420 PRINT AT 2,5;"THE MAVZOTH"
    "WARRIORS"
430 LET M$="V<A>"
440 FOR N=1 TO 4
450 FOR X=5 TO 28-N*5
460 POKE DF+X+132,CODE M$(N)
470 POKE DF+X+131,136
480 NEXT X
490 NEXT N

```

```

500 PRINT AT 6,0;"THEIR HEAT AS"
510 FOR N=0 TO 20
520 NEXT N
530 PRINT AT 9,8;"THE SPIRAL"
540 FOR N=5 TO 8
550 FOR X=5 TO 28-(N-4)*5
560 POKE DF+X+363, CODE M$(N)
570 POKE DF+X+362, 135
580 NEXT X
590 NEXT N
600 PRINT AT 13,10;"THE SPIRAL"
610 FOR N=5 TO 23
620 POKE DF+N+495, 151
630 POKE DF+N+494, 8
640 NEXT N
650 PRINT AT 17,4;"THE DISINTER"
660 FOR N=5 TO 23
670 POKE DF+N+627, 140
680 POKE DF+N+626, 135
690 NEXT N
700 PRINT AT 23,0;"PRESS A"
710 GOTO 710+NOT INKEY$=""
720 RAND USR SCREEN
730 PRINT AT 2,3;"THE OFFENDER"
740 PRINT AT 4,3;"ANTOMATTER CE"
750 PRINT AT 9,3;"THE RYELS"
760 PRINT AT 14,5;"THE RYELS"
770 GOTO 770+NOT INKEY$=""
780 CLS
790 RAND USR SCREEN
800 PRINT AT 8,3;"ENTER SAME"
810 SLOW
820 LET Y$=INKEY$
830 REM
840 GOTO 820+30*(CODE Y$)=CODE
"1" AND CODE Y$=CODE "9")
850 LET B=VAL Y$
860 POKE 16910, B
870 POKE 17938, B*10
880 PRINT AT 23,0;"PRESS A"
890 GOTO 890+(INKEY$="")
900 GOTO 900+(INKEY$<>"")
910 REM
920 RAND USR SCREEN
930 PRINT AT 23,0;"SCORE= 00000"
940 BASES REMAINING= 10
950 RAND USR GRAPHICS
960 FOR N=0 TO PEEK 16910+3
970 NEXT N
980 RAND USR START
990 IF PEEK (DF+791)=28 THEN GO
TO 1020
1000 RAND USR SCREEN
1010 GOTO 940
1020 REM
1030 LET U=DF+767
1040 LET G$=CHR$ (PEEK U+128)+CH
R$ (PEEK (U+1)+128)+CHR$ (PEEK (
U+2)+128)+CHR$ (PEEK (U+3)+128)+
CHR$ (PEEK (U+4)+128)
1050 CLS
1060 RAND USR SCREEN
1070 PRINT AT 3,11;"GAME OVER";A
T 8,5;"PRESS A"
1080 PRINT AT 3,7;"PRESS A"
1090 LET Y$=INKEY$
1100 GOTO 1070-290*(Y$="Y")+20*(
Y$="N")
1110 REM
1120 CLS
1130 FOR N=0 TO 30
1140 POKE 16918, N
1150 RAND USR END
1160 NEXT N
1170 STOP
1180 SAVE "ARENA2"
1190 SLOW
1200 RUN
1210 REM
1220 REM
1230 REM
1240 REM
1250 REM
1260 REM
1270 REM
1280 REM
1290 REM
1300 REM
1310 REM
1320 REM
1330 REM
1340 REM
1350 REM
1360 REM
1370 REM
1380 REM
1390 REM
1400 REM
1410 REM
1420 REM
1430 REM
1440 REM
1450 REM
1460 REM
1470 REM
1480 REM
1490 REM
1500 REM
1510 REM
1520 REM
1530 REM
1540 REM
1550 REM
1560 REM
1570 REM
1580 REM
1590 REM
1600 REM
1610 REM
1620 REM
1630 REM
1640 REM
1650 REM
1660 REM
1670 REM
1680 REM
1690 REM
1700 REM
1710 REM
1720 REM
1730 REM
1740 REM
1750 REM
1760 REM
1770 REM
1780 REM
1790 REM
1800 REM
1810 REM
1820 REM
1830 REM
1840 REM
1850 REM
1860 REM
1870 REM
1880 REM
1890 REM
1900 REM
1910 REM
1920 REM
1930 REM
1940 REM
1950 REM
1960 REM
1970 REM
1980 REM
1990 REM

```

Oliger DOS v.2 & M-SCRIPT

Kent Cook

I just received my JLO SAFE Disk Basic V2.11 EPROM and it's GRR-REAT. Here's how I changed the BASIC in my A&J Microdrive version of Dohay/MSCRIPT to use it.

Load MSCRIPT and exit to BASIC. Enter the new lines 6 and 7. Add "GO SUB VAL "500" to the end of line 600 and delete line 605. Enter the new BASIC from line 610 to line 750. Leave lines 990 and 991, but delete lines 995 and 996.

Run the program and make a fresh SAVE by following the prompts. If a tape or disk error ever stops the program, simply RUN to get back into it. As an incidental benefit, the BASIC program is now 44 bytes shorter. If you are opposed to tape storage or need more room, you could delete the "TAPE or OLIGER" subroutine, all references to it, the expressions conditional upon the variable named dos, and all lines relating to tape operations. Tape backup could still be done in the direct command mode from BASIC.

```

6 IF dos THEN LOAD "/MSCRIPT"
7 IF NOT dos THEN LOAD "CODE"

800 CLS : PRINT TAB VAL "8";"
SAVE LOAD "1: SAVE ENTIRE
TEXT" : SAVE TEXT "BLOCK" :
SAVE MSCRIPT" : 4: LOAD NEW TEX
T" : 5: APPEND TEXT" : 0: RETURN
GO SUB VAL "500"
610 INPUT "IF K$<"0" OR K$<"
OR LEN K$>1 THEN GO TO VAL "6
612 IF K$="0" THEN GO TO VAL "5
614 IF K$<"3" THEN INPUT "FILE
TO VAL "5" AND K$<"3"
616 TO VAL "5"
618 TO VAL "5"
620 NOT LN THEN GO TO VAL "5
622 GO SUB VAL "750": IF dos TH
EN NOT dos THEN SAVE F$CODE
GO SUB VAL "750"
624 TO VAL "5"
626 NOT B1 OR NOT B2 THEN GO
SUB VAL "750": IF dos TH
EN NOT dos THEN SAVE F$CODE
628 N1+B2-B1: GO SUB VAL "750"
630 TO VAL "5"
632 GO SUB VAL "750": IF dos TH
EN NOT dos THEN SAVE F$CODE
634 SAVE "/MSCRIPT" LINE VAL "5":
VAL "5"
636 IF NOT dos THEN SAVE "MSCRI
PT" LINE VAL "5": SAVE "MSCRIPT"
CODE VAL "5500" VAL "5500": CLS
PRINT AT VAL "10": "REWIND an
d PLAY to VERIFY" VERIFY "" VE
RIFY ""
638 GO TO VAL "50"
640 GO SUB VAL "750": IF dos TH
EN NOT dos THEN LOAD F$CODE
642 IF F$<"Y" AND K$<"Y" THEN RETU
RN
644 GO TO VAL "50"
646 GO SUB VAL "550": IF dos TH
EN NOT dos THEN LOAD F$CODE
648 IF F$<"Y" AND K$<"Y" THEN RETU
RN
650 IF F THEN POKE VAL "23606",
VAL "150": POKE VAL "23607", VAL
"242": RETURN
651 POKE VAL "23606", VAL "0": P
OKE VAL "23607", VAL "50": RETURN

```

CUSTOMIZE YOUR SIGN DESIGNER

by Chia-Chi Chao

Most, if not all, of you should have heard about the famous Print Shop program for computers such as Apple and IBM. It is a graphics package which allows you to create cards, signs, and banners for various occasions. Well, how about our Timex Sinclair computers? Zebra Systems came to the rescue by putting out a series of programs similar to the Print Shop: Card Designer, Sign Designer, and Banner Designer. They are priced at \$19.95 each, and they all have received good reviews.

I purchased a copy of Sign Designer some time ago, thinking that I could also create "small signs" to make cards. I found out that is not the case. Therefore, I decided to modify the program to shrink those full page signs.

Note: This conversion was written for and tested on Citizen MSP-10 (EPSON compatible) printer, but it should work for most other printers as well. Since I can not test this modification on other printers, you may find the result different from what is described here.

A pixel on the screen is converted to four pixels on the printout, doubled horizontally and vertically. In order to make a card, you don't need that magnification. By examining the BASIC and machine language parts of Sign Designer, the printing routine was found and deciphered.

```
C1BE CALL C20D ;get bit pattern into register A
C1C1 PUSH AF ;save data on stack
C1C2 CALL C25E ;send to printer
C1C5 POP AF ;get it back
C1C6 PUSH AF ;send to printer again,
C1C7 CALL C25E ; to be
C1CA POP AF ; taken out
C1CB LD HL,BFC9 ;test for single/double density
C1CE BIT 0,(HL) ; bit set if double density
C1D0 JR Z,C1DA ; skip if single density
C1D2 PUSH AF ;send data to printer,
C1D3 CALL C25E ; to be
C1D6 POP AF ; deleted
C1D7 CALL C25E ;send to printer for the last time
C1DA rest of the routine
```

It is easy to reduce the horizontal pixels because the same data is just sent to the printer twice in a row (four times for double density mode). This can be accomplished by poking some zeros in the printing routine to take out the extra instructions.

```
XXOOXXOO      XO XO
XXOOXXOO  -> converted to -> XO XO
OOXXOOXX      XO XO
OOXXOOXX      XO XO
```

For vertical pixels, a MC routine has to be added to process these data. Here is the simplified algorithm:

1. Calculate first pair of coordinates.
2. Get pattern #1.
3. Filter out unnecessary bits.
4. Calculate second pair of coordinates.
5. Get pattern #2.
6. Filter out unnecessary bits.
7. Combine the two patterns.
8. Send data to printer.

```

XOXO
#1  XOXO -> filter -> XOXO
    OXOX          OXOX
    OXOX
                                -> combine ->
                                XOXO
                                OXOX
                                OXOX
                                XOXO
#2  OXOX
    OXOX -> filter -> OXOX
    XOXO          XOXO
    XOXO

```

In order to create a card, two pictures should appear at opposite corners of a page, one normal and one upside down so that when the paper is folded, both designs are right-side-up. The above algorithm only shrinks the sign, so a further modification is needed to flip the picture. Suppose we have a 3 by 5 array:

	0	1	2	3	4		4	3	2	1	0	
0	A	B	C	D	E	==>	2	O	N	M	L	K
1	F	G	H	I	J		1	J	I	H	G	F
2	K	L	M	N	O		0	E	D	C	B	A

When the original program wants to print "A" in position (0,0), we actually want to print "O" in position (2,4). So, if (row,column) are the coordinates, we need (2-row,4-column) to reverse the array. In the Sign Designer, there are 356 rows and 216 columns stored in the computer, so we only need to change steps 1 and 4 of the algorithm a little to flip the picture.

Now let's start modifying the Sign Designer. First, load the machine language part:

```
CLEAR 45739
LOAD ""CODE 45928,19606
```

We need to poke some numbers to change and delete some MC instructions.

```
POKE 49599,173 ;B2AD (CALL ENTRY)
POKE 49600,178 ;branch to our modification

POKE 49656,8 ;(LD BC,0008)

POKE 49601,0 ;delete the extra
POKE 49602,0 ;printing instructions
POKE 49603,0
POKE 49604,0
POKE 49605,0
POKE 49618,0
POKE 49619,0
POKE 49620,0
POKE 49621,0
POKE 49622,0
```

Then use a BASIC loader or an appropriate editor such as Hot-Z to enter the machine code routines. There are 174 bytes, so be patient.

```
10 FOR X=45741 TO 45914
20 PRINT X;" ";
30 INPUT A: PRINT A
40 POKE X,A: NEXT X
```

Values to input using the above loader

Address	Dec	Hex	Address	Dec	Hex	Address	Dec	Hex
45741	71	47	45806	197	C5	45871	246	F6
45742	58	3A	45807	213	D5	45872	4	04
45743	172	AC	45808	229	E5	45873	203	CB
45744	178	B2	45809	237	ED	45874	99	63
45745	254	FE	45810	91	5B	45875	40	28
45746	0	00	45811	116	74	45876	2	02
45747	120	78	45812	191	BF	45877	246	F6
45748	40	28	45813	33	21	45878	2	02
45749	56	38	45814	215	D7	45879	203	CB
45750	205	CD	45815	0	00	45880	115	73
45751	13	0D	45816	167	A7	45881	40	28
45752	194	C2	45817	237	ED	45882	2	02
45753	197	C5	45818	82	52	45883	246	F6
45754	213	D5	45819	34	22	45884	1	01
45755	229	E5	45820	91	5B	45885	201	C9
45756	205	CD	45821	179	B3	45886	14	0E
45757	225	E1	45822	237	ED	45887	0	00
45758	178	B2	45823	91	5B	45888	205	CD
45759	245	F5	45824	118	76	45889	98	62
45760	237	ED	45825	191	BF	45890	192	C0
45761	91	5B	45826	33	21	45891	237	ED
45762	118	76	45827	100	64	45892	75	4B
45763	191	BF	45828	1	01	45893	91	5B
45764	19	13	45829	167	A7	45894	179	B3
45765	19	13	45830	237	ED	45895	121	79
45766	19	13	45831	82	52	45896	203	CB
45767	19	13	45832	34	22	45897	56	38
45768	237	ED	45833	93	5D	45898	203	CB
45769	75	4B	45834	179	B3	45899	25	19
45770	116	74	45835	235	EB	45900	203	CB
45771	191	BF	45836	205	CD	45901	56	38
45772	237	ED	45837	62	3E	45902	203	CB
45773	67	43	45838	179	B3	45903	25	19
45774	91	5B	45839	205	CD	45904	203	CB
45775	179	B3	45840	35	23	45905	56	38
45776	205	CD	45841	179	B3	45906	203	CB
45777	62	3E	45842	245	F5	45907	25	19
45778	179	B3	45843	237	ED	45908	9	09
45779	205	CD	45844	91	5B	45909	230	E6
45780	225	E1	45845	93	5D	45910	7	07
45781	178	B2	45846	179	B3	45911	205	CD
45782	79	4F	45847	27	1B	45912	13	0D
45783	241	F1	45848	27	1B	45913	194	C2
45784	7	07	45849	27	1B	45914	201	C9
45785	7	07	45850	27	1B			
45786	7	07	45851	205	CD			
45787	7	07	45852	62	3E			
45788	177	B1	45853	179	B3			
45789	225	E1	45854	205	CD			
45790	209	D1	45855	35	23			
45791	193	C1	45856	179	B3			
45792	201	C9	45857	24	18			
45793	87	57	45858	179	B3			
45794	230	E6	45859	95	5F			
45795	96	60	45860	175	AF			
45796	15	0F	45861	203	CB			
45797	15	0F	45862	67	43			
45798	15	0F	45863	40	28			
45799	95	5F	45864	2	02			
45800	122	7A	45865	246	F6			
45801	230	E6	45866	8	08			
45802	6	06	45867	203	CB			
45803	15	0F	45868	83	53			
45804	179	B3	45869	40	28			
45805	201	C9	45870	2	02			

After you are done, save the MC portion on tape:

SAVE "CardDesign"CODE 45740,19794.

For more information on the MC routines, please see the commented disassembly listing. Now load the BASIC part of Sign Designer: LOAD ". You can BREAK it after the MC part starts to load. Warning! Do NOT use RUN, or you will erase lots of important data. Certain lines have to be added/deleted to support our added features:

```
2114 [ENTER]
2202 [ENTER]
```

```
12 LOAD "CardDesign"CODE 45740,19794
2006 OVER 0:PRINT AT 9,0;" (64 spaces) ":OVER 1
2120 INPUT "T)op or B)ottom ? ";k$
2125 IF k$="t" OR k$="T" THEN POKE 45740,0: POKE
49133,1: GO TO 2140
2130 IF k$="b" OR k$="B" THEN POKE 45740,1: POKE
49133,43: GO TO 2140
2135 GO TO 2120
2140 POKE 49109,176: POKE 49110,1: POKE 49149,18
2150 RANDOMIZE USR 49535
9990 SAVE "CardDesign" LINE 12
9992 SAVE "CardDesign"CODE 45740,19794
```

After you finish changing the BASIC portion, load the MC part back: LOAD "CODE. Then use GOTO 9990 to save the complete program (BASIC and MC).

You may also want to add this line to your regular Sign Designer: 2113 POKE 49149,36. Location 49149 controls the top and bottom margins, in 1/72 of an inch. The original value is 32, but the value I calculated is 36, at least for my printers. If you find that the second sign you print consecutively is higher on the page than the first one, then you need to change that value. For our modification to print cards, that value is 18 (half of 36), as done in line 2140.

Location 49133 (in lines 2125 and 2130) controls the left margin, in characters. The values of 1 and 43 are for EPSON (or compatible) printers, assuming the paper is centered correctly. You may need to adjust these values to get the best result on your printer.

You can go through the BASIC program and change every "Sign" in PRINT statements to "Card", but it does take a lot of work. You use the "Card" Designer as usual. Just pick Top and Bottom portions correctly when printing your creations. Top is upside-down at upper left corner, and bottom is normal at lower right corner. If you have any questions about this conversion, please send SASE to: Chia-Chi Chao, 73 Sullivan Drive, Moraga, CA. 94556-1209. Enjoy!



```

HORIZN EQU BF74h ;horizontal counter
VERTCL EQU BF76h ;vertical counter
HRZMAX EQU 215 ;216 columns (0-215)
VRTMAX EQU 356 ;356 rows (0-355)
TOPFLG EQU 0 ;flag for top portion
BTMFLG EQU 1 ;flag for bottom part

B2AC FORMAT DEFB BTMFLG ;code for top or bottom
B2AD ENTRY LD B,A ;save A in B
B2AE LD A,(FORMAT) ;load A with format
B2B1 CP TOPFLG ;which function?
B2B3 LD A,B ;get original A back
B2B4 JR Z,TOP ;goto proper function

B2B6 BOTTOM CALL C20Dh ;the stolen instruction
B2B9 PUSH BC ;save registers
B2BA PUSH DE ; on stack
B2BB PUSH HL ;
B2BC CALL FILTER ;get the bits wanted
B2BF PUSH AF ;save 1st ptrn on stack
B2C0 SECOND LD DE,(VERTCL);get vertical coord.
B2C4 INC DE ;increase 4 to get
B2C5 INC DE ; coordinate for next
B2C6 INC DE ; row
B2C7 INC DE ;
B2C8 LD BC,(HORIZN);get horizontal coord.
B2CC LD (NEWHRZ),BC;save it for later
B2D0 CALL SUB1 ;get bit pattern
B2D3 CALL FILTER ;again, get bits wanted
B2D6 COMBIN LD C,A ;save 2nd pattern in C
B2D7 POP AF ;get first pattern back
B2D8 RLCA ;rotate first pattern
B2D9 ; 4 times
B2DA RLCA ;
B2DB RLCA ;
B2DC OR C ;combine two patterns
B2DD POP HL ;get the registers back
B2DE POP DE ;
B2DF POP BC ;
B2E0 RET ;return to orig routine

B2E1 FILTER LD D,A ;save orig pattern in D
B2E2 AND 96 ;get the upper two bits
B2E4 RRCA ;rotate right into
B2E5 RRCA ; position
B2E6 RRCA ;
B2E7 LD E,A ;store it in E
B2E8 LD A,D ;get orig pattern back
B2E9 AND 06 ;get the lower two bits
B2EB RRCA ;rotate into position
B2EC OR E ;combine the four bits
B2ED RET ;
B2EE TOP PUSH BC ;save registers
B2EF PUSH DE ; on stack
B2F0 PUSH HL ;
B2F1 INIT LD DE,(HORIZN);get horizontal coord.
B2F5 LD HL,HRZMAX ;highest horiz coord.

```

```

B2F8 AND A ;clear carry flag
B2F9 SBC HL,DE ;flip horizontal coord.
B2FB LD (NEWHRZ),HL;save it in memory
B2FE LD DE,(VERTCL);get vertical coord.
B302 LD HL,VRTMAX ;highest vert coord.
B305 AND A ;clear carry flag
B306 SBC HL,DE ;flip vertical coord.
B308 LD (NEWVRT),HL;save it in memory
B30B START EX DE,HL ;put HL (v-coord) in DE
B30C CALL SUB1 ;get first bit pattern
B30F CALL REVERS ;get bits and reverse
B312 PUSH AF ;save 1st ptrn on stack
B313 SECAND LD DE,(NEWVRT);get last vert. coord.
B317 DEC DE ;decrease 4 to get
B318 DEC DE ; coordinate for
B319 DEC DE ; next row
B31A DEC DE ;
B31B CALL SUB1 ;get second pattern
B31E CALL REVERS ;reverse and filter it
B321 JR COMBIN ;combine two patterns

B323 REVERS LD E,A ;put bit pattern in E
B324 XOR A ;initialize A=0
B325 BIT 0,E ;test each of the 4
B327 JR Z,NEXT1 ; bits and set
B329 OR 08 ; appropriate bits in A
B32B NEXT1 BIT 2,E ;
B32D JR Z,NEXT2 ;
B32F OR 04 ;
B331 NEXT2 BIT 4,E ;
B333 JR Z,NEXT3 ;
B335 OR 02 ;
B337 NEXT3 BIT 6,E ;
B339 JR Z,DONE ;
B33B OR 01 ;
B33D DONE RET ;

B33E SUB1 LD C,00 ;C has to be zero
B340 CALL C062h ;taken from original
B343 LD BC,(NEWHRZ); routine
B347 LD A,C ;
B348 SRL B ;
B34A RR C ;
B34C SRL B ;
B34E RR C ;
B350 SRL B ;
B352 RR C ;
B354 ADD HL,BC ;
B355 AND 07 ;
B357 CALL C20Dh ;
B35A RET ;

B35B NEWHRZ DEFW 0 ;storage for horizontal
B35D NEWVRT DEFW 0 ;and vertical coords

```

TYD☆BYTS

OS64 Bug

I recently discovered a latent catastrophe in my OS64 cartridge version 1.72. The cart. sets RAMTOP at 63255. This is ABOVE the machine stack and RAM-resident code. If a long program overwrites the machine stack, it's BYE-BYE!

The normal place for RAMTOP in double display file modes is 63255, the first byte below UDG. Changing it for double display files was not built into the TS2068 ROM and the OS64 cart. doesn't do it either.

A "CLEAR 63255" will, of course, reset RAMTOP to the usual and less dangerous address and permit the use of all UDGs. RAMTOP can be raised to 63423 at the expense of the UDGs.

ZXLR8 & TS150

Still having trouble with ZXLR8 and the TS1500? Try this: SPEED IT UP! That's right, try timing values between 7 and 10. This is because the high-pass filter in the TS1500, coupled with the transistor pre-conditioner, can make "1" bits (long pulses) appear as two short pulses. By speeding the routine up (shortening the pulses), this effect is eliminated. With the TS1500, the faster you go, the better it works (subject to the limitations of your recorder). In many cases, this is all it takes; you won't even need an inverting cable!

Kent Cook

BASIL'S COMPENDIUM

Basil Wentworth

Input Output PEEK & POKE

So far, the Z80 chip must impress you as being a singularly inflexible device. It will do what you ask it to, and do that very well--so long as you ask it to tell you the value of $B+256^{\circ}C$. And as long as you include enough information in the program to define B and C. And, once you have defined them, there seems to be no way to change their values, short of rewriting the program.

It seems a little like the remark ascribed to Henry Ford: people could have any color car they wanted--as long as they wanted black.

Well, there's no way to get around that ultimate inflexibility. We can tell the computer to do much more than compute BC, but it still won't do any more than we tell it to. (Although some times you can't be sure what you've told it to do, until it's done it.) Once we've written that program, we have committed ourselves. Just as in BASIC.

There's no reason, though, that we can't write a number of programs, and send the computer to one or another, as conditions demand. That's what the Sinclair people did when they designed the ZX81's ROM. Those 8K are packed with machine code routines which, taken all together, contain the entire repertoire of amazing things that your computer can do.

But there's still the question of communicating with your computer. In BASIC, you have INPUT. And you have PRINT (or LPRINT) for an output. And you can PRINT a lot more than one variable. Whereas in machine code...

Well, you just wrote a program that would compute the fifth power of 2, for instance. Wouldn't it be nice to be able to compute the sixth power of 2, without rewriting the whole program? Or the fifth power of 3? Or maybe to compute a whole series of the Nth power of 2 from N=1 to N=6, and see the series of answers?

INPUT and OUTPUT in machine code are accomplished by PEEK and POKE. You INPUT data, you store that data in the computer's memory, and tell the chip to PEEK it at the proper time. For an output, you tell the computer to POKE the information into the memory at some place you specify.

PEEK with Register A...

PEEK and POKE are identified in the mnemonic lists by parentheses. For instance, LD A,(HL) means to load register A with the data found at the address defined by the contents of HL. Notice that it doesn't mean to load A with the data held in HL itself--it couldn't, anyway, because HL contains two bytes, while A has room for only one.

LD A,(NN) is a little more complicated, but still perfectly logical. First of all, NN represents a

number that is made up of two bytes, Least Significant Byte first. This number, when it appears in parentheses, represents a memory address. And LD A,(NN) means to load register A with the data found in the memory byte at that address.

Let's try an example to help us to understand that one. Look at Figure 14-1. The data at 16515 and 16516 are 256-imal for 16513 (i. e., $129+256^{\circ}64$). Byte 16514 tells us to PEEK that location, and to LD A with the number it finds there. The rest of the program moves that number into BC, where we'll be able to read it.

```
16514  58 U      LD A,(16513)
16515  129 Gr Q
16516  64 RND
16517   6 Gr T   LD B,0
16518   0 space
16519  79 ?     LD C,A
16520 201 TAN   RET
```

PRINTUSR 16514 gives
PEEK 16513=234

Figure 14-1. LD A,(NN)

Before you go any further, what is that number? If you can't figure it out, try PRINT PEEK 16513.

Now PRINTUSR 16514, and check your prediction. You should get 234, the CODE for "REM".

```
LD A,      (BC) (DE) (HL) (NN)
           10   26   126* 58
```

* NOTE that 126 is a minor
no-no (see Chapter 7)

```
LD(HL),    A      B      C      D
           119   112   113   114
```

```
LD(HL),    E      H      L      N
           115   116   117   54
```

```
LD B,(HL)   70
LD C,(HL)   78
LD D,(HL)   86
LD E,(HL)   94
LD BC,(NN)  *75
LD DE,(NN)  *91
LD HL,(NN)  42 or *107**
```

*NOTE: preceded by 237

**NOTE: mistake in ZX81 manual

```
LD(NN),    A      BC      DE      HL
           50   *67   *83   34 or
                                   *99
```

*NOTE: preceded by 237

Table 14-1. Some PEEKs and POKEs

Try another. POKE 16515,130. What does PRINT USR 16514 give you now? Since you are asking to PEEK 16514, your output should be 58.

A collection of instructions involving PEEK and POKE is shown in Table 14-1.

...or 2-Byte Registers

You will notice that there are instructions in the table for loading BC, DE, or HL with (NN)--that is, with the data in the byte at memory location $N1+256*N2$. However, you will remember that register pairs must be loaded with two bytes of data. Accordingly, a PEEK on their behalf must PEEK two consecutive bytes. We only have to specify the address of the first of the two.

So LD HL,(NN) tells the computer:

1. Compute the address represented by $N1+256*N2$, where N1 is the first of the NN pair, and N2 is the second.
2. Load H with the data in that address.
3. Load L with the data found in the very next byte.

So let's have a go at it. Follow Figure 14-2. PRINT USR 16514 should give you:

PEEK 16513+256*PEEK 16514, or

$234+256*42=10986$

```
16514  42 E    LD HL,(16513)
16515  129 Gr Q
16516   64 RND
16517   68 ?    LD B,H
16518   77 ?    LD C,L
16519  201 TAN  RET
```

```
PRINT USR 16514 gives
PEEK 16513+256*PEEK 16514
=234+256*42=10986
```

Figure 14-2. LD HL,(NN)

Oddly enough, there is also a two-byte instruction, 237:107, that has the same meaning as 42. Even more oddly, my ZX81 handbook has it mislabeled as LD DE,(NN). It should be LD HL,(NN). (The TS 2068 handbook has it right.)

Try again, with LD DE,(NN) (=237:91), and then again with LD BC,(NN) (=237:75). In the DE case, you'll have to replace bytes 16517 and 16518 with LD B,D (=86) and LD C,E (=75). (Not necessarily in that order, by the way--it doesn't matter which you do first, as long as you do both.) And when you LD BC,(NN) those two bytes can be deleted.

NOP

Or they can be replaced with zeroes. A zero instruction is a do-nothing. Its mnemonic is NOP (no operation). It is sometimes used to delay a program for a few microseconds (like the PAUSE in BASIC, but with much, much shorter delays) while events in the real world catch up. If you want a longer delay, you can insert a loop of NOP's, just as you might use

```
100 FOR X=1 TO 17
110 NEXT F
```

in BASIC. Except for the delay, which you will never notice in these elementary applications, the NOP has no effect whatsoever.

You can use it in practice sessions to reserve a byte or a number of bytes, which will be given significant numbers later on. Or to null out a byte of instruction that you no longer want. Just be sure that the byte you're replacing with zero actually IS AN INSTRUCTION. If you replace a data byte with a zero, the computer will take the zero as bona fide data, and may give you rather unforeseen results.

One other caution (two, actually). If you null out an instruction, be sure that you also null out all the data bytes associated with it. And if you are zeroing out a two-byte instruction, be sure that you replace BOTH bytes with zeroes. If you violate either of these rules, the computer will be almost sure to misinterpret the bytes that you neglected to null out.

POKE

In the POKE case, the destination address is sure to be greater than 255. After all, the bytes at addresses lower than 256 are all in ROM, and can't be POKEd. Therefore, all addresses you will ever POKE will be expressed as two bytes. For example, your POKE instruction might take the form of

```
LD(NN),A    or
LD(NN),L
```

In the first case, the contents of register A (one byte, less than 256) will be POKEd into the address given by NN (=N1+256*N2). If you are POKeing the contents of a two-byte register, then the contents of the low-byte register (L, C, or E) will be POKEd into the memory at address N1+256*N2, and the contents of the high-byter will be POKEd into the following byte of memory. In other words, data storage will follow the pattern that you are used to: Least Significant Byte first.

```
16514   1 Gr 1    LD BC,69
16515   69 ?
16516   0 space
16517  237 GOSUB LD(16522),BC
16518   67 ?
16519  138 Gr F
16520   64 RND
16521  201 TAN    RET
16522   27 .     null
16523   27 .     null
```

```
After RAND USR 16514, PRINT
(PEEK 16522+256*PEEK 16523)
should give 69
```

Figure 14-3. LD(NN),BC

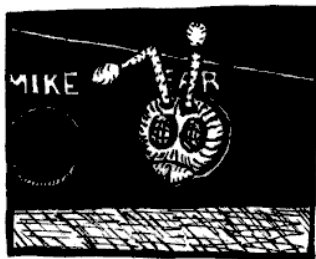
CLASSIFIED

Do you have something to sell?
Let everyone know! Just \$2.75 a
line will put it here. 32 column
(screen) format, include spaces
in your text, and send CK or MO
with your listing.

Send to SWN Classified, 602 S.
Mill St., Louisville, OH 44641

TS-1000; TS-2040; MEMOTECH ADD-ONS
(RAMS, K-BOARD, HRG, PRINTER I/F, ETC);
BROTHER THERMAL PRINTER; 12" MONO-
CHROME MONITOR; SOFTWARE, BOOKS (WILL
SEND LIST ON REQUEST). A. LAVIOLETTE
16-1385 BERNARD MONTREAL CANADA H2V 1W1

FOR SALE: TS1000 FEMALE EDGE CONNECTORS
(WITH SLOT) \$4 EACH T. WOODS, BOX 64,
JEFFERSON, NH 03583



```
16514 62 Y LD A,69
16515 69 ?
16516 50 M LD(16520),A
16517 136 Gr H
16518 64 RND
16519 201 TAN RET
16520 27 . null
16521 27 . null
```

After RAND USR 16514, PRINT
PEEK 16520 should give 69

Figure 14-4. LD(NN),A

For practice, try the exercises in Figures 14-3 to
14-5. Remember to provide nulls at the end of the
REM statement to furnish a destination for the
POKed data. In these cases, don't PRINT USR 16514,
as you did before. Use RAND USR 16514, followed by
the PRINT statement given with each exercise.

That's all for now. There will be more.

SyncWare News

CONTENTS

FOR YOUR SUPPORT	3
FORUM	4
2K-SDP: Serial Data Port	
Kent Cook	9
TS1000 MYSTERY PROGRAM II	
Gregory Harder	17
RUN MSCRIPT w/OLIGER DOS	
Kent Cook	18
CUSTOMIZE YOUR SIGN DESIGNER	
Chia-Chi Chao	19
TYDBYTS	8, 21
BASIL'S COMPENDIUM	
Basil Wentworth	22

FIRST-CLASS MAIL
U.S. POSTAGE
PAID
Jefferson, NH
Permit No. 1

95633

Editor:
Jeffrey Moore
602 S. Mill Street
Louisville, OH 44641

Technical Director:
Fred Nachbaur
C-12, Mtn. Stn. Group Box
Nelson, BC V1L 5P1 Canada

Advertising, Subscriptions:
Thomas B. Woods
P.O. Box 64
Jefferson, NH 03583

Submissions, announcements, letters to the editor,
and all other material of editorial interest should
be sent to the Ohio address.

Copyright 1987 SyncWare News

Subscribe ! \$ 16.95

For Canada and Mexico please add \$3.00 postage.

1 year (6 issues)